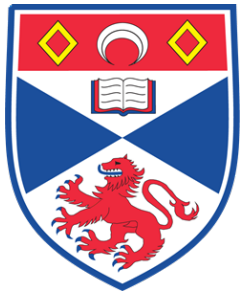
University of St Andrews

CS5014 Coursework 1

Machine Learning

Author:
150008022

February 22, 2019



Goal

The goal of this practical was to cleanse and process real world data in order to produce a regression model, and evaluate its performance. [1]

Part I
Loading and Cleaning the Data

Numpy was used to load the csv file. The header row was skipped, and to ensure there were no missing values, the invalid raise flag was also used when parsing the data. This would raise an exception if any rows were found to be missing data.

The input and output columns were separated into two variables x and y, as in accordance with the notation used in lectures.

Part II
Analysing and Visualising the Data

Firstly, a histogram of each of the input variables and outputs were plotted in order to visualise the distribution of the values. When comparing to the histograms from the given paper TODO fig 1) ref paper, most of the plots matched. The difference was identified to be caused by 10 bins always being used (the default if not specified by numpy.hist), and the paper would sometimes use more. However, it was still clear that none of the variables had a gaussian distribution.

In order to identify which variables affected the outputs most, scatter graphs were made between each input and output variable. Feature scaling was used to allow for comparisons between values that could have very different ranges. Mean normalisation was used for the scatter plots. The resulting plots resembled those from the given paper (Figure 2 from given paper TODO), with different x-axis values since mean normalisation gave

values in the range $-0.5 \leq x \leq 0.5$.

Part III
Feature Selection

To try and identify which features had the strongest effect on the outputs, both the Pearson and Spearman rank correlation coefficients were considered. It was noted that the Pearson correlation would give a perfect value when the two variables were linearly related, whilst the Spearman correlation (a similiar alternative, and the one used in the original paper) would give a perfect value when the variables were monotonically related. Pearson is meant for use with continuous variables, whilst Spearman can be used for both continuous and ordinal variables, which our data set contains. Given these factors, the Spearman rank correlation coefficient was used as a filter method.

From the scatter plots, some variables appeared to have a possible linear relationship with the outputs (for example, X7 and Y1), whilst others had a monotonic relationship (for example, X1 and Y1). Using the Scipy *stats.spearmanr* method, the correlation coefficients were easily calculated, alongside a p-value, as shown in table 1. Immediately from this result we can see that X6

Based on these metrics, a number of features were to be chosen. Given the number of input features, using them all would have negative effects on the choosen algorithm. Including each feature could result in overfitting, and would require more training data to eliminate this issue. It would also increase the computation tie required to train our model. Features that have no real effect on the outputs would also act simply as noise, which would negatively effect our model.

Recursive Feature Elimination was also considered to help choose which features were important.

Table 1: Spearman rank correlation coefficients, with p values

X	Y	Rho	p
1	1	0.62	0.00
1	2	0.65	0.00
2	1	-0.62	0.00
2	2	-0.65	0.00
3	1	0.47	0.00
3	2	0.42	0.00
4	1	-0.80	0.00
4	2	-0.80	0.00
5	1	0.86	0.00
5	2	0.86	0.00
6	1	-0.00	0.91
6	2	0.02	0.63
7	1	0.32	0.00
7	2	0.29	0.00
8	1	0.07	0.06
8	2	0.05	0.20

Conclusion

References

[1] Facebook. create-react-app.